Performance benchmarks for a next generation numerical dynamo model

Hiroaki Matsui, Eric M. Heien, Julien Aubert, Jonathan Aurnou, Margaret Avery, Ben Brown, Bruce A Buffett, Friedrich Busse, Ulrich R Christensen, Christopher J Davies, et al.

AGU PUBLICATIONS

Geochemistry, Geophysics, Geosystems

RESEARCH ARTICLE
10.1002/2015GC006159

Performance benchmarks for a next generation numerical dynamo model

Hiroaki Matsui[1,2], Eric Heien[1,2], Julien Aubert[3], Jonathan M. Aurnou[1,4], Margaret Avery[5], Ben Brown[1,6], Bruce A. Buffett[1,7], Friedrich Busse[8], Ulrich R. Christensen[9], Christopher J. Davies[5], Nicholas Featherstone[1,10], Thomas Gastine[9], Gary A. Glatzmaier[1,11], David Gubbins[12], Jean-Luc Guermond[13], Yoshi-Yuki Hayashi[14], Rainer Hollerbach[15], Lorraine J. Hwang[1,2], Andrew Jackson[16], Chris A. Jones[15], Weiyuan Jiang[17], Louise H. Kellogg[1,2], Weijia Kuang[17], Maylis Landeau[18], Philippe Marti[10], Peter Olson[1,18], Adolfo Ribeiro[4], Youhei Sasaki[19], Nathanaël Schaeffer[20], Radostin D. Simitev[21], Andrey Sheyko[16], Luis Silva[21], Sabine Stanley[1,22], Futoshi Takahashi[23], Shin-ichi Takehiro[24], Johannes Wicht[9], and Ashley P. Willis[25]

[1]Computational Infrastructure for Geodynamics Dynamo Working Group, University of California, Davis, California, USA, [2]Department of Earth and Planetary Sciences, University of California, Davis, California, USA, [3]Institut de Physique du Globe de Paris, Sorbonne Paris Cité, Université Paris-Diderot, CNRS, Paris, France, [4]Department of Earth and Space Sciences, University of California, Los Angeles, California, USA, [5]Scripps Institution of Oceanography, University of California, San Diego, La Jolla, California, USA, [6]Department of Astrophysical and Planetary Sciences, University of Colorado, Colorado, USA, [7]Department of Earth and Planetary Science, University of California, Berkeley, California, USA, [8]Institute of Physics, University of Bayreuth, Bayreuth, Germany, [9]Max Planck Institute für Sonnensystemforschung, Göttingen, Germany, [10]Department of Applied Mathematics, University of Colorado, Boulder, Colorado, USA, [11]Department of Earth and Planetary Sciences, University of California, Santa Cruz, California, USA, [12]School of Earth Environment, University of Leeds, Leeds, UK, [13]Department of Mathematics, Texas A&M University, College Station, Texas, USA, [14]Center for Planetary Science/Department of Earth and Planetary Sciences, Kobe University, Kobe, Japan, [15]Department of Applied Mathematics, University of Leeds, Leeds, UK, [16]Institute of Geophysics, ETH Zurich, Zurich, Switzerland, [17]Planetary Geodynamics Laboratory, NASA Goddard Space Flight Center, Greenbelt, Maryland, USA, [18]Department of Earth and Planetary Sciences, Johns Hopkins University, Baltimore, Maryland, USA, [19]Department of Mathematics, Kyoto University, Kyoto, Japan, [20]Université de Grenoble Alpes, CNRS, ISTerre, Grenoble, France, [21]School of Mathematics and Statistics, University of Glasgow, Glasgow, UK, [22]Department of Physics, University of Toronto, Toronto, Canada, [23]Department of Earth and Planetary Sciences, Kyushu University, Fukuoka, Japan, [24]Research Institute for Mathematical Sciences, Kyoto University, Kyoto, Japan, [25]School of Mathematics and Statistics, University of Sheffield, Sheffield, UK

Key Points:
- Performance benchmark tests for 15 dynamo codes up to 16,384 processor cores
- 2-D-parallelized or 3-D-parallelized codes should keep scale well to millions of processor cores
- 2-D-parallelized spherical harmonic expansion method is the crusial for future dynamo model

Correspondence to:
H. Matsui,
hrmatsui@ucdavis.edu

Citation:
Matsui, H., et al. (2016), Performance benchmarks for a next generation numerical dynamo model, *Geochem. Geophys. Geosyst.*, *17*, 1586–1607, doi:10.1002/2015GC006159.

Received 27 OCT 2015
Accepted 9 FEB 2016
Accepted article online 12 FEB 2016
Published online 6 MAY 2016

Abstract Numerical simulations of the geodynamo have successfully represented many observable characteristics of the geomagnetic field, yielding insight into the fundamental processes that generate magnetic fields in the Earth's core. Because of limited spatial resolution, however, the diffusivities in numerical dynamo models are much larger than those in the Earth's core, and consequently, questions remain about how realistic these models are. The typical strategy used to address this issue has been to continue to increase the resolution of these quasi-laminar models with increasing computational resources, thus pushing them toward more realistic parameter regimes. We assess which methods are most promising for the next generation of supercomputers, which will offer access to $O(10^6)$ processor cores for large problems. Here we report performance and accuracy benchmarks from 15 dynamo codes that employ a range of numerical and parallelization methods. Computational performance is assessed on the basis of weak and strong scaling behavior up to 16,384 processor cores. Extrapolations of our weak-scaling results indicate that dynamo codes that employ two-dimensional or three-dimensional domain decompositions can perform efficiently on up to $\sim 10^6$ processor cores, paving the way for more realistic simulations in the next model generation.

1. Introduction

The Earth's magnetic field is generated by convection in the liquid iron alloy of the Earth's outer core. Understanding the origin and evolution of the Earth's magnetic field and the dynamics of the outer core are long-standing grand challenges in geophysics that have been approached with a variety of theoretical,

observational, laboratory, and computational methods. Numerical simulations have played a central role in this quest, particularly since *Glatzmaier and Roberts* [1995]. These challenges also exist in numerical studies of convective dynamos in other planets [*Stanley and Glatzmaier*, 2010] and in stars [*Glatzmaier*, 1984; *Featherstone and Hindman*, 2016; *Hotta*, 2016]. Nevertheless, these simulations have yet to answer some fundamental questions about how the geodynamo actually works, because of the extreme resolution that is required to reach fully Earth-like behavior in dynamo models. In particular, it is anticipated that the flow in the outer core exhibits a vast range of length scales, ranging from the thickness of viscous boundary layers ($\sim$0.1 m) to the diameter of the core ($\sim 7\times10^6$ m), plus a commensurately full range of time scales, which present generation models cannot capture. Researchers currently deal with this challenge by adopting values for viscosity and thermal diffusivity that are much larger than those expected in the Earth's core. Achieving Earth-like values is unlikely in the foreseeable future, but by pushing the parameters toward more extreme values we hope to realize solutions that are dynamical similar with the appropriate balance of forces.

In order to access these more extreme parameter regimes, the next generation of numerical dynamos will require much improved spatial and temporal resolution, and consequently, efficient, massively parallel computational capability is a must. However, most existing dynamo codes involve data structures and solution methods for which efficient parallelization is problematic, the best example being the traditional spherical harmonic representations of the fluid velocity, magnetic field, and other dependent variables as described in section A1.2. Following the method described in *Orszag and Patterson* [1972] for periodic boundary conditions, the advection, Lorentz force, magnetic induction, and in some codes, the Coriolis force, are evaluated in physical space using spherical harmonic transforms, while the time integration and linear terms are solved in spectral space for each spherical harmonic mode. Consequently, spherical harmonic transforms must be used to transform the nonlinear terms back to spectral space, making this the most time consuming step in geodynamo simulations, and the most difficult to efficiently parallelize. For small-sized and mid-sized dynamo simulations, deviations from ideal parallel efficiency are often not the primary concern, but they become the limiting factor for the massive-scale simulations which will be used for the next generation dynamo models.

To meet this challenge, we have carried out performance and accuracy tests on 15 widely used numerical dynamo models. We chose two standard, well-studied benchmarks for our tests. Our first benchmark is the same as dynamo benchmark 1 in *Christensen et al.* [2001, 2009], which uses vacuum (i.e., electrically insulating) magnetic field conditions on the outer and inner boundaries of the spherical shell. This vacuum boundary condition is local in spherical harmonic space but global in physical space, and therefore it poses a challenge for codes that are based on local physical-space methods [*Chan et al.*, 2001; *Matsui and Okuda*, 2004b]. Because of this, we have added a second benchmark that uses so-called pseudo-insulating (-vacuum) magnetic boundary conditions, as described in *Jackson et al.* [2014]. These boundary conditions are local in physical space, although they are less realistic than the insulating conditions for most planetary applications.

Briefly, each benchmark consists of solving an initial value problem for convection and magnetic field generation in a rotating, electrically conducting spherical shell. Exact definitions of our two benchmarks, including the full set of governing equations, dimensionless input parameters and output variables, run diagnostics, plus boundary and initial conditions, are given in Appendix A, along with the standard solutions we use to measure code accuracy.

Identical tests were performed on a common platform on all 15 codes, representing the 14 participating individuals or teams in Table 1. Of these codes, 13 use spherical harmonic transforms, while the remaining 2 use local methods. Among the codes using spherical transforms, nine use so-called transpose methods, in which the parallelization direction alternates between the linear calculations and the spherical transforms. For the two spherical harmonic codes (Parody and XSHELLS), parallelization is in the radial direction throughout the simulation. SPmodel is parallelized in θ direction for the spherical harmonic transform, and H2000 is not parallelized. Calypso and Rayleigh are parallelized in two directions for each calculation (2-D parallelization), and the other spherical harmonic expansion codes are parallelized in one direction for each calculation (1-D parallelization). SFEMaNS is parallelized three dimensionally because the finite element mesh in SFEMaNS can be decomposed in either direction in a meridional plane, and the Fourier transform used in the longitudinal direction is also parallelized. Summaries of the numerical schemes used in each of the participating codes are given in section A4.

Table 1. List of Participated Codes[a]

Code	Version	Author	MPI	OpenMP
Spherical Harmonic Expansion With Chebyshev Collocation Method				
MagIC	3 and 5	Johannes Wicht, Thomas Gastine, and Ulrich Christensen	1-D TP	Yes
Rayleigh		Nicholas Featherstone	2-D TP	No
SBS	1.6.0	Radostin Simitev, Friedrich Busse, and Luis Silva	1-D TP	No
SPmodel	0.8.0	Youhei Sasaki, Shin-ichi Takehiro, and Yoshi-Yuki Hayashi	1-D θ	Yes
UCSC code	R2-p0	Gary A. Glatzmaier	1-D TP	No
Spherical Harmonic Expansion With Radial Compact FDM				
SpF-MoSST		Weijia Kuang and Weiyuan Jiang	1-D TP	No
TITECH code		Futoshi Takahashi	1-D TP	Yes
Spherical Harmonic Expansion With Radial FDM				
Calypso	Dev.	Hiroaki Matsui	2-D TP	Yes
ETH code	1.2.1	Philippe Marti, Andrey Sheyko, and Andy Jackson	1-D TP	No
H2000		Rainer Hollerbach	No	No
LSD code	14 Mar 2014	Ashley Willis, Christopher Davies, Margaret Avery, Chris Jones, and David Gubbins	2-D TP	No
PARODY-JA	2.6	Maylis Landeau, Julien Aubert	1-D *r*	Yes
XSHELLS	Dev.	Nathanaël Schaeffer	1-D *r*	Yes
Code	**Version**	**Author and Methods**	**MPI**	**OpenMP**
Local Methods				
GeoFEM	12 Mar 2012	Author: Hiroaki Matsui Method: Finite element method (FEM)	3-D	Yes
SFEMaNS	Ver. 2.2	Author: Jean-Luc Guermond, Adolfo Ribeiro, and Francky Luddens Methods: quadratic FEM in meridional section and Fourier expansion for ϕ direction	3-D	No

[a]Dev.: development version (see detailed description); *r*: parallelized in radial direction; θ-NL: parallelized in θ direction for nonlinear terms; TP: transpose method.

A primary motivating factor for the present study is to explore which numerical methods might be most useful on a "petascale computer," which can carry out one quadrillion floating point operations per second (i.e., 10^{15} FLOPS, equivalent to one petaflops, or PFLOPS). To that end, we have conducted a rather ambitious extrapolation of our weak-scaling results to 10^6 processor cores in section 2.1. While a "petascale-sized problem" ($\ell_{max} \geq 4096$, where ℓ_{max} is the truncation level of the spherical harmonics) would be most ideal for testing the capabilities of these different codes, the resources currently available, combined with the fact that many existing codes have yet to optimize for the memory and communication issues associated with such large problems, means that our test problems are much more modest in size. A side effect is that a substantial drop-off in parallel efficiency is observed for many codes at processor core counts of 10^3 and 10^4 in the strong scaling test because the test problems are relatively small. For larger problem sizes, however, the point at which this drop-off occurs is pushed to larger core counts, and we expect our weak-scaling results to remain valid.

2. Performance Tests

Performance benchmarks measure the speed and parallel capability of the codes in a single computational environment. This study uses TACC Stampede consisting of 6400 computational nodes, each node configured with two Intel 8-core Xeon E5 series processors. The computational nodes are connected with InfiniBand. The total peak performance for the Xeon processors of Stampede is more than two PFLOPS, and network performance is 56 Gb/s [*Texas Advanced Computing Center*, 2013]. Each node also has Xeon Phi coprocessor which has more than seven PFLOPS of peak performance in total, but we only use the main Xeon processors in the present benchmark. We use Intel compiler version 14.0 and Intel MPI Ver. 4.1.3 to build executable processes. (Performance also depends on the compiler. For example, the elapsed time for MagIC5 compiled by Intel compiler Ver. 15.0 with 2048 processor cores is 0.48 times of Intel compiler version 14.0 in Appendix B.)

We conducted two different tests of code performance, termed weak and strong scaling, respectively. To standardize our performance tests, we measure the elapsed time per time step averaged over 100 time steps from the initial condition and excluding data IO and initialization (e.g., LU decomposition for the linear calculation). Although some of the codes feature variable time stepping, all of our tests were conducted with a fixed time step.

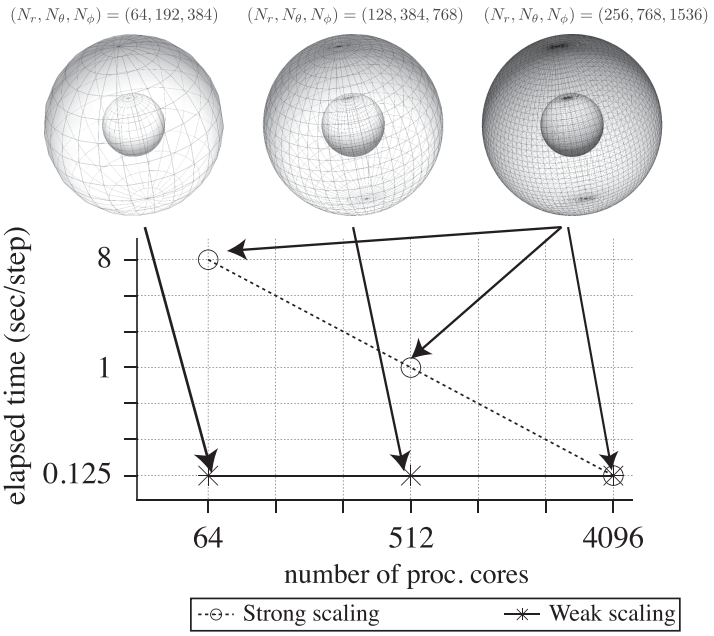

Figure 1. Schematic diagram illustrating the difference between strong and weak-scaling tests. In the weak-scaling test, the number of degrees of freedom (DOFs; proportional to grid size) increases in proportion to the number of processor cores; ideal weak scaling corresponds to constant elapsed time. In the strong scaling test, the DOFs is held constant, so with ideal strong scaling the elapsed time decreases linearly with the number of processor cores.

2.1. Weak Scaling

In a weak-scaling test, a sequence of runs is performed using an increasing number of processor cores, keeping the problem size on each core approximately constant, while measuring the elapsed time per time step. In our tests, we fix the problem definition, but vary the resolution, increasing the number of grid points in proportion to the number of processor cores, so that the number of degrees of freedom (DOFs) for each scalar on each processor remains approximately constant from one run to the next. We adjusted the DOFs on each core so that the elapsed time per time step is between 0.1 and 1.0 s. For reference, approximately 10^7 time steps are needed for one magnetic diffusion time at an Ekman number of $E=1\times10^{-5}$ and at a magnetic Prandtl number $Pm = 0.5$, so the imposed elapsed time per step is required to ensure practical calculations. To account for the differences in accuracy among the spherical harmonic codes in Appendix C, we increase the radial resolution of the finite difference models. We use 2 times and 4 times more radial DOFs in the compact finite difference method (CFDM) and finite difference method (FDM) codes compared with the Chebyshev codes, respectively. By comparison we run SFEMaNS with a total DOFs comparable to that used in Rayleigh. For the specific case of 4096 processor cores, SFEMaNS has slightly fewer DOFs (i.e., 92%). Thus, the spherical harmonic codes have similar accuracy, whereas SFEMaNS is performed with somewhat lower accuracy. The precise definition of DOFs we use is given in Appendix B.

As illustrated in Figure 1, ideal weak scaling defined this way corresponds to constant elapsed time per time step, independent of the number of processor cores used. Said another way, ideal weak scaling corresponds to elapsed time being independent of the total DOFs in the calculation. Codes that use spherical transforms often perform poorly in this test because the number of computations for the Legendre transform in the spherical transform grows like $O(\ell_{max}^3)$. In the present weak-scaling test, ℓ_{max} increases with $O(N_{Core}^{1/3})$, where N_{Core} is the number of processor cores. Consequently, ideal weak scaling for the Legendre transform is $O(N_{Core}^{1/3})$. Good performance in weak scaling corresponds to a small slope in Figure 1.

Weak-scaling results for three codes that use two-dimensional or three-dimensional domain decomposition are plotted by marks in Figure 2a. Rayleigh and Calypso use spherical harmonic transforms, whereas SFEMaNS uses locally based finite elements in the meridional plane and Fourier transforms in the longitudinal direction. We fit the elapsed times shown in Figure 2a to power laws of the form

$$t = AN_{Core}^{p}, \tag{1}$$

in which the exponent p measures the departures from ideal weak scaling. In particular, if the Legendre transform controls elapsed time, then the predicted exponent is $p = 1/3$. Although the absolute value of the elapsed time is different in Figure 2a, all three codes have exponents below $p = 1/3$ and the local-based code SFEMaNS has a p value much less than 1/3. Rayleigh and Calypso need less computation time than SFEMaNS up to 2×10^6 processor cores because Rayleigh and Calypso have smaller power law coefficients A than that for SFEMaNS. A value less than 1/3 for Rayleigh and Calypso implies that additional factors, such as communication and memory bandwidth, have a significant impact on the total computation time.

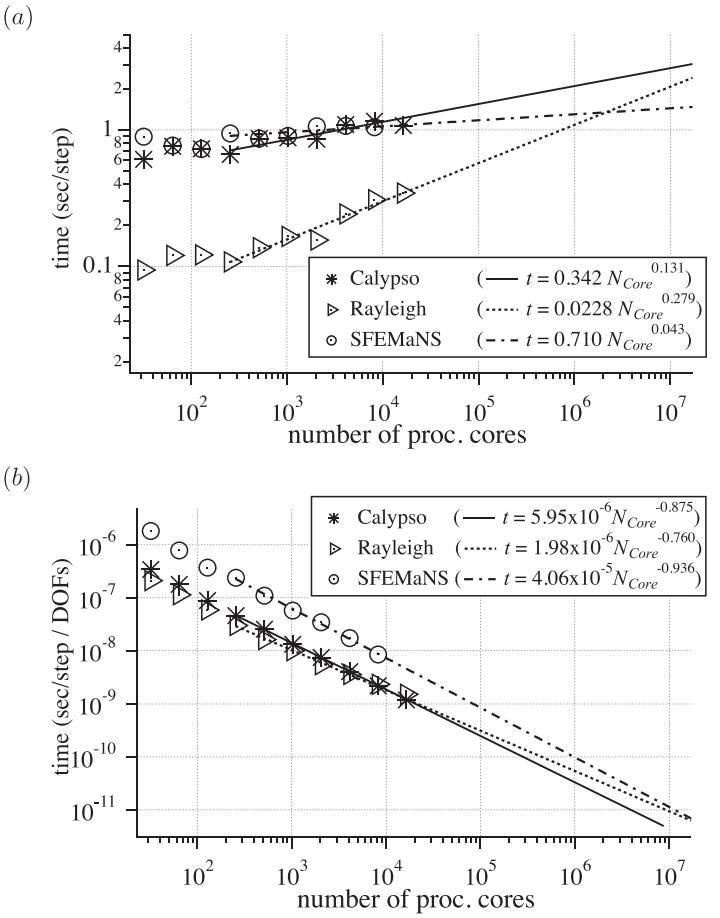

Figure 2. Weak-scaling test: (a) wall clock time per time step versus number of processor cores for three codes with two-dimensional or three-dimensional domain decomposition described in section A4. The wall clock time per step normalized by the total DOFs is plotted in Figure 2b. Lines are power law fits of elapsed time t in seconds/step extrapolated to a million processor cores. Spatial resolutions for 4096 processor cores case are 6.7×10^7, 2.8×10^8, and 6.2×10^7 DOFs for Rayleigh, Calypso, and SFEMaNS, respectively.

We performed a weak-scaling test on all the codes except H2000, and derived the best fitting p values shown in Figure 3. Compared to Rayleigh and Calypso, larger deviations from ideal weak scaling are evident in the spherical harmonic codes that use only 1-D domain decompositions. The general result is around $p = 1/3$ for codes based on spherical harmonic expansion. Parody has $p \simeq 0.9$ due to the increased expense of its linear solver relative to the other codes. SBS-DRS and SPmodel also have $p \simeq 1$.

Our weak-scaling results lead to several general conclusions. First, codes using a two or more dimensional domain decomposition have a large parallelization capability. In some cases, according to our extrapolations, such codes may retain enough efficiency to use on platforms with millions or even tens of millions of processor cores. Second, parallelization of the transpose methods applied to the spherical transform has a greater effect on the elapsed time compared to parallelization in the radial direction, because time for the linear calculation increases faster than the Legendre transform in the radial parallelization. Finally, the code XSHELLS, with MPI in the radial direction and OpenMP in the angular direction, also performed well in our scaling tests. Although more limited than 2-D MPI domain decomposition, this approach may also be interesting with future computers that feature greater numbers of cores per node.

What about our goal to extrapolate to millions of processor cores? The weak-scaling results in Appendix B include only four data points for most of the codes, and it is meaningless to extrapolate such small samples that far. In order to extrapolate our weak scaling to a million processor cores, we additionally tested the performance of three particularly promising codes, Rayleigh, Calypso and SFEMaNS, using up to 16,384 processor cores, then fit the results to a power law with exponent p. As the results in Figure 2a show, extrapolating the two spherical harmonic codes Rayleigh and Calypso achieves approximately 1 and 2 seconds per time step at $2^{21} \simeq 2.1\times10^6$ processor cores, respectively. In 2^{21} core case, spatial resolution for Rayleigh and Calypso would be $(N_r, \ell_{max}) = (2048, 4095)$ and (8192, 4095), respectively. SFEMaNS would also achieve a time step between 1 and 2 seconds per time step at 2^{21} processor cores with a slightly lower number of DOFs than that for Rayleigh.

One way to more effectively compare the results is to normalize the computation time by the number of DOFs (see Figure 2b). In this representation, the exponents of the scaling and extrapolation change. For example, Calypso scales with an exponent of $p = -0.875$ which is smaller than the ideal scaling for the Legendre transform (see below), whereas Rayleigh scales $p = -0.760$ but with a smaller A value. The

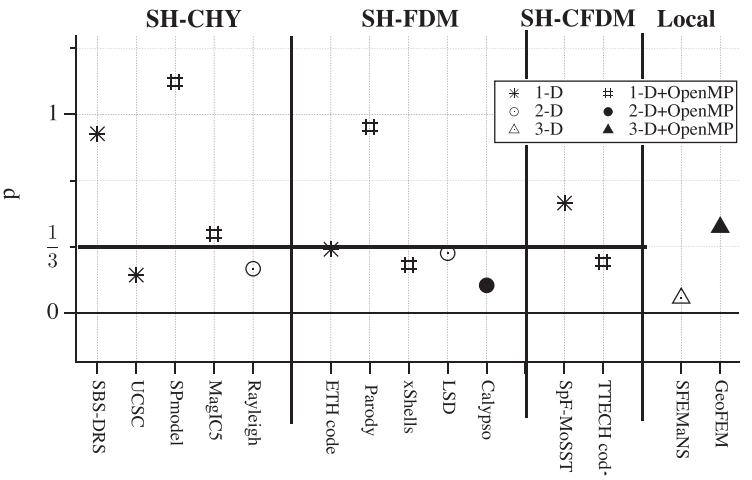

Figure 3. Fitted exponent p of the elapsed time as a function of N_{core} for weak-scaling test. Ideal weak scaling for Legendre transform ($p = 1/3$) is shown by the thick line. The result for code H2000 is not shown because H2000 is not parallelized. Symbol type denotes the number of directions of MPI parallelization and OpenMP parallelization.

SFEMaNS exponent is $p = -0.935$, much smaller than both Calypso and Rayleigh but with a larger A value. The extrapolated elapsed times for SFEMaNS and Rayleigh converge approximately at 1×10^7 processor cores. However, as described in Appendix C, codes using spherical harmonics expansion need less DOFs than that for local methods. Ectrapolating the present results, Rayleigh will require much less computational time than SFEMANS to obtain comparable accuracy for 1×10^7 processor cores.

It is fair to ask if our extrapolation to millions of processor cores is actually meaningful. In Figure 2a, the ideal exponent for the Legendre transform is $p = 1/3$, which is larger than the exponents of Calypso and Rayleigh. The small exponents p for Calypso and Rayleigh might simply reflect the communication time, which has the ideal exponent of $p = 0$ when bandwidth, not latency, is the limiter, as we expect for these problem sizes. However, communication time may increase faster with larger parallelization level [*Engelmann*, 2014]. We evaluate the exponent p for the communication time in Figure 4. The exponent $p = 0.129$ for Calypso is almost the same as our previous estimate for the total elapsed time ($p = 0.131$) and smaller than the ideal scaling for the Legendre transform $p = 1/3$. And, the exponent p for Rayleigh is almost same as the ideal scaling for the Legendre transform. We might expect the exponent to approach $p = 1/3$ with increasing parallelization level and spatial resolution for Rayleigh and Calypso, confirming that our scaling does not reflect the communication time.

2.2. Strong Scaling

In a strong scaling test, a sequence of runs is performed using an increasing number of processor cores, while fixing the total problem size. In our test, we ensure fixed problem size by fixing the grid and the spherical harmonic truncation, so that the DOFs per core decrease in proportion to the number of processor

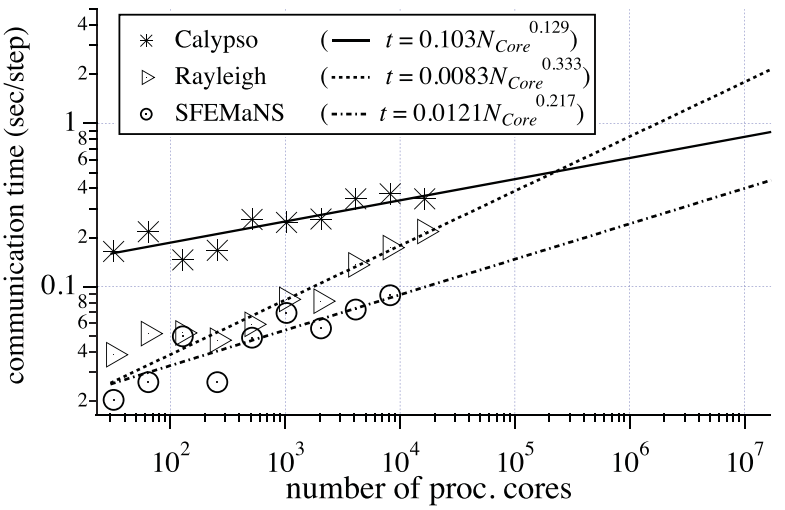

Figure 4. The data communication time per step in the weak-scaling test for the three codes shown in Figure 2. Lines are power law fits of elapsed time t in seconds/step extrapolated to a million processor cores.

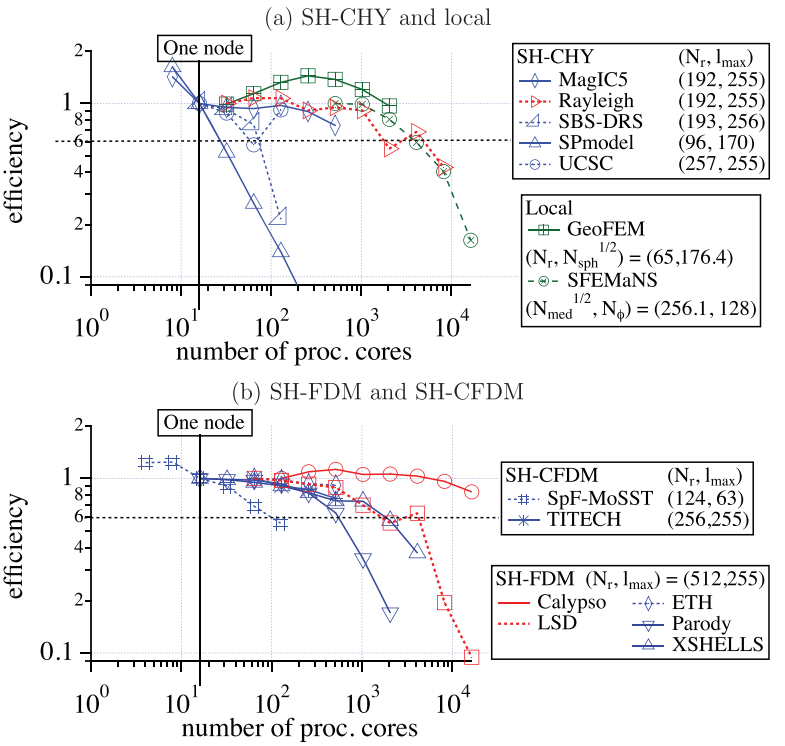

Figure 5. Parallel efficiency ϵ as a function of the number of processor cores in the strong scaling. The efficiency is referred by the result using 16 processor cores (1 node). Results for codes using spherical harmonic-Chebyshev and local methods are plotted in Figure 5a, and results for spherical harmonic-FDM and spherical harmonic-CFDM are plotted in Figure 5b. Codes with 1-D parallelization are plotted in blue, codes with 2-D parallelization are plotted in red, and the 3-D-parallelized model is plotted in green. A solid line is used for codes with hybrid parallelization (MPI and OpenMP), and a dotted line is used for the other codes. $\epsilon = 1$ is the ideal scaling.

cores used. With ideal strong scaling, according to this prescription, the trend of elapsed time in Figure 1 should have a slope of -1. Good performance in strong scaling is therefore characterized by a trend with slope close to -1, whereas poor performance is often characterized by a trend with slope near zero.

To quantify the deviations from ideal strong scaling, we define an efficiency factor ϵ as

$$\epsilon = \frac{t_{ref}}{t}\frac{N_{ref}}{N_{Core}}, \tag{2}$$

where t is the elapsed time, N_{Core} is the number of processor cores used, and t_{ref} and N_{ref} denote a reference time and number of processor cores. In this scaling test, $N_{ref} = 16$, which corresponds to a single Stampede node. According to (2), $\epsilon = 1$ for ideal strong scaling.

Figure 5 shows ϵ versus N_{ref} for all 15 codes except for H2000, grouped by spherical harmonic-Chebyshev and local methods (Figure 5a) and spherical harmonic-finite difference methods (Figure 5b). Efficient strong scaling is difficult to maintain for this problem size with 1-D MPI parallelization beyond $N_{ref} \sim 10^3$, and better results are found with 2-D or 3-D MPI parallelization, which maintain high efficiency up to $N_{ref} \sim 10^4$ in some cases.

To further quantify this effect, we define the parallelization limit, the number of processor cores N_{ref} where ϵ falls to 0.6 in strong scaling. Figure 6 shows this limit for all 15 codes. Most of the 1-D decomposition codes without OpenMP fall below our efficiency threshold around $2^6 = 64$ processor cores in this test. The ETH code keeps good scaling up to 512 processor cores, which is the parallelization limit of the ETH code in the present strong scaling test. The parallelization limit increases for some spherical harmonic codes with OpenMP, but this is still far below from what is needed for the next generation dynamo code. More encouragingly, the codes with multidimensional domain decomposition plus OpenMP fare better, remaining within the parallelization limit using $2^{13} = 8192$ processor cores and for three codes (Rayleigh, LSD, and Calypso), using the largest allowance of Stampede: $2^{14} = 16{,}384$ processor cores.

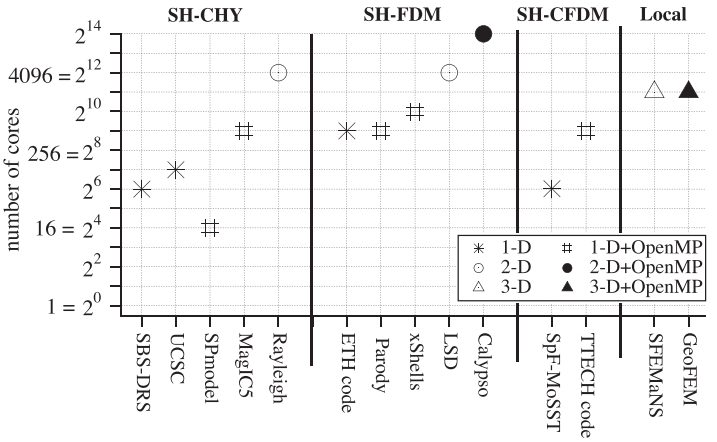

Figure 6. Maximum number of processor cores for the strong scaling to keep parallel efficiency $\epsilon > 0.60$ in the strong scaling test. Symbol type denotes the number of directions of MPI parallelization and OpenMP parallelization.

Another question addressed by this test is whether hybrid (MPI and OpenMP) parallelization is faster than full MPI parallelization. We tested Calypso with hybrid parallelization and full MPI parallelization and found that the elapsed times for both are comparable (see Table B5 and Figure B2).

Besides the scaling capability, our performance tests also show large differences in raw performance between codes at a given core count. Indeed, from the fastest to the slowest, a factor of 20 to 100 is observed (see Figures B1 and B2). The fastest code in the SH-FDM category is, by far XSHELLS, while in the SH-CHY category MagIC and Rayleigh are very close. We also note that the ETH, LSD, and TITECH codes use corrector iterations on the nonlinear terms, doubling the work per step over the majority of codes using Adams-Bashforth methods.

Local methods are more suitable for a massively parallel computational environment because they only require data communications for overlapping areas among subdomains. However, the insulating magnetic boundary condition cannot be described locally, hence is only applied in few models (*Matsui and Okuda* [2005] by finite element method (FEM); *Chan et al*. [2007] by FEM; *Wicht et al*. [2009] by finite volume method (FVM), and SFEMaNS by FEM in meridional plane and Fourier transform in zonal direction *Ribeiro et al*. [2015]). If an iterative linear solver is used, several communication steps are required to satisfy the insulating boundary condition for the magnetic field and the solenoidal boundary condition for the velocity field. Consequently, the linear solver becomes the most time consuming part for the local methods.

3. Conclusions

We performed performance and accuracy benchmark tests for geodynamo simulations using 15 codes with up to 16,384 processor cores on TACC Stampede. The results of the performance benchmark show that codes using 2-D or 3-D parallelization models have the capability to run efficiently with up to 16,384 processor cores. The parallelization limit for these codes using 2-D or 3-D parallelization is estimated to be 10^7 processor cores, and elapsed time for these codes is expected to be approximately 1 s/step by using 10^6 processor cores. In the weak-scaling test, growth of the elapsed time for SFEMaNS (a local method) is the smallest with increasing the parallelization. The elapsed times for Calypso and Rayleigh, which use spherical harmonic expansion, grow with less than the ideal scaling for the Legendre transform. The elapsed times for SFEMaNS and Rayleigh are projected to converge at approximately 1×10^7 processor cores for problems with comparable DOFs. However, the accuracy benchmark tests show that finite difference methods require three times more DOFs in the radial direction compared to Chebyshev expansion method. Similar refinement would be required for local methods in other directions. Consequently, local methods will need longer elapsed times than spectral methods to achieve the same accuracy with the same number of processors. According to our results, spherical harmonic expansion methods with 2-D parallelization offer the best assurance of efficiency for geodynamo simulations that employ 10^6 to 10^7 processor cores.

We also observed that hybrid parallelization (MPI + OpenMP) increases the parallelization level. However, one test using MPI and hybrid parallelization in the same code produced comparable results when the number of processor cores was the same. Radial parallelization is the fastest with less than 1000 processor cores for the problem sizes tested here, but this approach imposes severe restrictions on the radial grid when the number of MPI parallelizations greatly exceeds the desired number of radial levels. In addition, run time for the codes with a 1-D, radial parallelization increases rapidly due to the waiting time in the linear solver for band matrices, except for XSHELLS which reasonably mitigates the waiting time in the linear solver.

The next generation of geodynamo simulations with more realistic parameters will be performed on a petascale computer. To realize the capabilities of a petascale computer, 2-D or 3-D MPI parallelization or MPI-OpenMP hybrid parallelization will be necessary. Based on the accuracy and performance benchmark tests in our study, the spherical harmonic expansion with 2-D parallelization method offers the best discretization for the dynamo simulations in a rotating spherical shell.

There are some questions that remain unresolved. First, the rate of convergence of the solution with increasing resolution in the radial direction is unknown. In order to determine this convergence rate, it would be necessary to compare solutions from each code using different resolutions in the radial as well as in the horizontal directions, a task that is beyond the scope of this study. Second, many codes include variable time stepping, but we fixed the length of the time step in order to simplify comparisons of the codes. It also remains an open question as to which methods of variable time stepping are most efficient and most accurate. A third unanswered question is whether the present accuracy benchmark results with laminar flows will carry over to solutions with more turbulent motions expected in the Earth's outer core. To determine code accuracy under these turbulent conditions, it will be necessary to define new benchmark problems, featuring more turbulent and time dependent flows.

Appendix A: Models

A1. Benchmark Definitions

A1.1. Governing Equations

The present benchmark is a magnetohydrodynamics (MHD) simulation in a rotating spherical shell modeled on the Earth's outer core. We consider a spherical shell from the inner core boundary (ICB) $r = r_i$ to the core mantle boundary (CMB) $r = r_o$ in a rotating frame which rotates at a constant angular velocity $\boldsymbol{\Omega}=\Omega\hat{z}$. The shell is filled with a Boussinesq fluid with constant diffusivities (kinematic viscosity ν, magnetic diffusivity η, and thermal diffusivity κ) and thermal expansion coefficient α. The inner core ($0 < r < r_i$) is corotating with the mantle, and assumed to be an electrical insulator or pseudo-insulating (-vacuum). The region outside the core is also assumed to be an electrical insulator or pseudo-insulating.

The nondimensional governing equations of the MHD dynamo problem are the following:

$$E\left[\frac{\partial \boldsymbol{u}}{\partial t}+(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}\right]=-\nabla P+E\nabla^2 u-2(\hat{z}\times\boldsymbol{u}) +RaT\frac{\boldsymbol{r}}{r_o}+\frac{1}{Pm}(\nabla\times\boldsymbol{B})\times\boldsymbol{B}, \tag{A1}$$

$$\frac{\partial T}{\partial t}+(\boldsymbol{u}\cdot\nabla)T=\frac{1}{Pr}\nabla^2 T, \tag{A2}$$

$$\frac{\partial \boldsymbol{B}}{\partial t}=\frac{1}{Pm}\nabla^2\boldsymbol{B}+\nabla\times(\boldsymbol{u}\times\boldsymbol{B}), \tag{A3}$$

and

$$\nabla\cdot\boldsymbol{u}=\nabla\cdot\boldsymbol{B}=0, \tag{A4}$$

where $\boldsymbol{u}$, P, $\boldsymbol{B}$, and T are the velocity, pressure, magnetic field, and temperature, respectively. To obtain the nondimensional equations, as given above, the shell width of $L=r_o-r_i$ and a viscous diffusion time of L^2/ν are selected as the length and time scales, respectively. The ratio of the inner core radius to the outer core radius is set to $r_i/r_o = 0.35$. Thus, inner core radius and outer core radius are $r_i = 7/13$ and $r_o = 20/13$, respectively. The magnetic field is nondimensionalized by the factor $B_0^2=\rho\mu_0\eta\Omega$, where ρ and μ_0 are the density of the fluid and the magnetic permeability, respectively. The temperature is normalized by the temperature difference between the inner and outer boundary of the shell ΔT. Gravity in equation (A1) is radially inward and varies linearly with radius according to $g=g_o r/r_o$, where g_o is amplitude of the gravity at the CMB.

A1.2. Spherical Harmonic Expansion

Although various time integration schemes and radial discretization methods are applied in numerical dynamo models for a rotating spherical shell (see section A4), most dynamo models use the spherical

harmonic expansion for the horizontal discretization for good accuracy and easy treatment of magnetic boundary condition. In spherical harmonic expansion, $\boldsymbol{u}$, P, $\boldsymbol{B}$, and T are expanded by a series of spherical harmonics. By using the triangular truncation, the temperature $T(r,\theta,\phi,t)$ and $P(r,\theta,\phi,t)$ are expanded by using the spherical harmonics $Y_l^m(\theta,\phi)$ with truncation level ℓ_{max} as

$$T(r,\theta,\phi,t)=\sum_{l=0}^{\ell_{max}}\sum_{m=-l}^{l}T_l^m(r,t)Y_l^m(\theta,\phi), \text{ and} \tag{A5}$$

$$P(r,\theta,\phi,t)=\sum_{l=0}^{\ell_{max}}\sum_{m=-l}^{l}P_l^m(r,t)Y_l^m(\theta,\phi), \tag{A6}$$

where $T_l^m(r,t)$ and $P_l^m(r,t)$ are the spherical harmonic coefficients of the temperature and pressure, respectively. Velocity and magnetic fields are expanded into the poloidal and toroidal components as

$$\boldsymbol{B}(r,\theta,\phi,t)=\sum_{l=1}^{\ell_{max}}\sum_{m=-l}^{l}\left[\nabla\times\nabla\times\left(B_{Sl}^m(r,t)Y_l^m(\theta,\phi)\hat{r}\right)+\nabla\times\left(B_{Tl}^m(r,t)Y_l^m(\theta,\phi)\hat{r}\right)\right], \text{and} \tag{A7}$$

$$\boldsymbol{u}(r,\theta,\phi,t)=\sum_{l=1}^{\ell_{max}}\sum_{m=-l}^{l}\left[\nabla\times\nabla\times\left(U_{Sl}^m(r,t)Y_l^m(\theta,\phi)\hat{r}\right)+\nabla\times\left(U_{Tl}^m(r,t)Y_l^m(\theta,\phi)\hat{r}\right)\right], \tag{A8}$$

where $\hat{r}$ is the unit radial vector. The scalar coefficients of the poloidal and toroidal components are denoted by suffix S and T, respectively. The scalar coefficients $T_l^m(r,t)$, $P_l^m(r,t)$, $B_{Sl}^m(r,t)$, $B_{Tl}^m(r,t)$, $U_{Sl}^m(r,t)$, and $U_{Tl}^m(r,t)$ are also discretized in the radial direction by Chebyshev expansion, finite difference method (FDM), or compact FDM and evolved by a time integration scheme. Following *Orszag and Patterson* [1972], fields are transformed into physical space to obtain the nonlinear terms (advection terms, Lorentz force, and magnetic induction), and nonlinear terms are transformed into scalar coefficients by spherical harmonic transform. Finally, the spherical harmonic coefficients of the fields are evolved by the linear terms and nonlinear terms.

A1.3. Control Parameters

The dimensionless numbers in the governing equations (A1)–(A4) are the Ekman number $E=\nu/\Omega L^2$; the modified Rayleigh number $Ra=\alpha g_o\Delta TL/\Omega\nu$; the Prandtl number $Pr=\nu/\kappa$; and the magnetic Prandtl number $Pm=\nu/\eta$. The following parameter values are used in the benchmarks:

$$\begin{aligned} E&=1.0\times10^{-3},\\ Ra&=100,\\ Pr&=1.0,\\ Pm&=5.0. \end{aligned} \tag{A9}$$

A1.4. Boundary Conditions for Velocity and Temperature

The boundary conditions exert a significant influence on the motion of the fluid and the overall dynamo process. Nonslip conditions for the velocity field and constant temperature values are imposed at both boundaries of the fluid shell for both cases as

$$\boldsymbol{u}=0 \text{ at } r=r_i \text{ and } r_o, \tag{A10}$$

$$T=1 \text{ at } r=r_i, \tag{A11}$$

$$T=0 \text{ at } r=r_o. \tag{A12}$$

A1.5. Magnetic Boundary Conditions

We consider the two benchmark cases with different magnetic boundary conditions in the present study. One is the insulating boundary case and another is the pseudo-insulating boundary case.

A1.6. Insulating Boundaries Case

In the insulating boundary case, the regions outside the fluid shell are assumed to be electrical insulators. This boundary condition is closer to the actual magnetic boundary conditions for the Earth's core. In the electric insulator, current density vanishes,

$$\boldsymbol{J}_{ext}=0, \tag{A13}$$

where the suffix "*ext*" indicates fields outside the fluid shell. At the boundaries of the fluid shell, the magnetic field $\boldsymbol{B}_{fluid}$, current density $\boldsymbol{J}_{fluid}$, and electric field $\boldsymbol{E}_{fluid}$ in the conductive fluid satisfy:

$$(\boldsymbol{B}_{fluid}-\boldsymbol{B}_{ext})=0, \tag{A14}$$

$$(\boldsymbol{J}_{fluid}-\boldsymbol{J}_{ext})\cdot\hat{r}=0, \tag{A15}$$

and

$$(\boldsymbol{E}_{fluid}-\boldsymbol{E}_{ext})\times\hat{r}=0.: \tag{A16}$$

Consequently, radial current density $\boldsymbol{J}$ vanishes at the boundary as

$$\boldsymbol{J}\cdot\hat{r}=0 \text{ at } r=r_i, r_o. \tag{A17}$$

In an electrical insulator, the magnetic field can be described as a potential field

$$\boldsymbol{B}_{ext}=-\nabla W_{ext}, \tag{A18}$$

where W_{ext} is the magnetic potential. The boundary conditions can be satisfied by connecting the magnetic field in the fluid shell at boundaries to the potential fields. By using the spherical harmonic expansion, the boundary condition can be expressed for each spherical harmonic coefficient by

$$\frac{l+1}{r}B_{Sl}^{m}(r)-\frac{\partial B_{Sl}^{m}}{\partial r}=B_{Tl}^{m}(r)=0 \text{ at } r=r_i, \text{ and} \tag{A19}$$

$$\frac{l}{r}B_{Sl}^{m}(r)+\frac{\partial B_{Sl}^{m}}{\partial r}=B_{Tl}^{m}(r)=0 \text{ at } r=r_o. \tag{A20}$$

A1.7. Pseudo-Insulating Case

Under the pseudo-insulating boundary condition, the magnetic field has only a radial component at the boundaries [e.g., *Harder and Hansen*, 2005]. This boundary condition can be applied easily to models using local methods since the boundary conditions can be defined at each grid point. Given the conservation of the magnetic field (A4), the magnetic boundary condition is

$$\frac{\partial}{\partial r}\left(r^2B_r\right)=B_\theta=B_\phi=0 \text{ at } r=r_i, r_o. \tag{A21}$$

By using the spherical harmonic expansion, the boundary condition can be expressed for each spherical harmonic coefficient by

$$\frac{\partial B_{Sl}^{m}}{\partial r}=B_{Tl}^{m}(r)=0 \text{ at } r=r_i, r_o. \tag{A22}$$

A2. Initial Conditions

A2.1. Initial Conditions for Velocity and Temperature

The initial velocity and temperature fields are defined by *Christensen et al*. [2001, 2009] for both the insulating and the pseudo-insulating benchmarks. Initially, there is no flow:

$$\boldsymbol{u}=0. \tag{A23}$$

Thermal convection is initiated by the following temperature perturbation:

$$T=\frac{r_o r_i}{r}-r_i+\frac{21}{\sqrt{17920\pi}}\left(1-3x^2+3x^4-x^6\right)\sin^4\theta\cos 4\phi, \tag{A24}$$

where $x=2r-r_i-r_o$.

A2.2. Initial Magnetic Field for Insulating Boundaries Case

Initial values are important for these benchmark tests. In particular, simulations must be started with a strong magnetic field to sustain the magnetic field in this parameter regime.

The initial magnetic field for the insulating magnetic boundaries case is defined in *Christensen et al*. [2001, 2009] as

$$B_r=\frac{5}{8}\left(8r_o-6r-2\frac{r_i^4}{r^3}\right)\cos\theta, \tag{A25}$$

$$B_\theta=-\frac{5}{8}\left(8r_o-9r+\frac{r_i^4}{r^3}\right)\sin\theta, \tag{A26}$$

and

$$B_\phi=5\sin\left(\pi(r-r_i)\right)\sin 2\theta. \tag{A27}$$

A2.3. Initial Magnetic Field for Pseudo-Insulating Boundaries Case

The initial magnetic field for the pseudo-insulating benchmark is defined by *Jackson et al.* [2014] as

$$B_r = \frac{5}{8}\frac{-48r_i r_o + (4r_o + r_i(4+3r_o))6r - (16+12(r_i+r_o))r^2 + 9r^3}{r}\cos\theta, \tag{A28}$$

$$B_\theta = -\frac{15}{4}\frac{(r-r_i)(r-r_o)(3r-4)}{r}\sin\theta, \tag{A29}$$

and

$$B_\phi = \frac{15}{8}\sin(\pi(r-r_i))\sin 2\theta. \tag{A30}$$

A3. Diagnostics and Data Outputs

Six data values are requested once a given dynamo simulation has reached a quasi steady state. These are

1. The kinetic energy averaged over the fluid shell E_{kin} defined by

$$E_{kin} = \frac{1}{V}\int \frac{1}{2}u^2 dV, \tag{A31}$$

   where V is the volume of the fluid shell.

2. The magnetic energy averaged over the fluid shell E_{mag} defined by

$$E_{mag} = \frac{1}{V}\frac{1}{EPm}\int \frac{1}{2}B^2 dV. \tag{A32}$$

3. Angular drift velocity of the field patterns in the zonal direction, ω.
4. Local temperature T.
5. Local zonal velocity u_ϕ.
6. Local θ-component of the magnetic field B_θ.

The local values (4)–(6) should be measured at a specific point though the solution propagates in the longitudinal direction. The local values are measured at middepth of the fluid shell ($r=(r_o+r_i)/2$) in the equatorial plane ($\theta=\pi/2$), with a ϕ-coordinate given by the conditions $u_r = 0$ and $\partial u_r/\partial\phi > 0$.

A4. Participating Models

MagIC (Ver. 3 and 5). MagIC uses the pseudo-spectral method. Spectral representations use Chebyshev polynomials in radius and spherical harmonics. Following *Glatzmaier* [1984], a second-order Adams-Bashforth scheme for the nonlinear terms and the Coriolis force and a Crank-Nicolson scheme for the remaining linear terms are used [*Christensen and Wicht*, 2007]. MagIC3 is parallelized in the radial direction only using OpenMP. MagIC5 is parallelized by MPI for spherical harmonic modes for linear operations and decomposed in the radial direction for nonlinear operations. MagIC is available at https://magic-sph.github.io/as a free software that can be used, modified, and redistributed under the terms of the GNU GPL v3 license.

Rayleigh. Spherical harmonics and Chebyshev polynomials are used to expand the variables in the horizontal and radial directions, respectively. Semiimplicit time integration is used (Crank-Nicolson for linear terms and Adams-Bashforth for the other terms). MPI parallelization is done in two dimensions. The linear operation is parallelized in spherical harmonic modes, the Legendre transform is parallelized in radial and harmonic degree m, and the Fourier transform and nonlinear calculations are parallelized in radial and meridional directions.

SPmodel. Spherical harmonics and Chebyshev polynomials are used to expand the variables in the horizontal and radial directions, respectively. The nonlinear terms and the Coriolis terms are evaluated in physical space and converted back to spectral space (so-called transform method). Diffusion terms are integrated by Crank-Nicolson scheme and the other terms are integrated by Adams-Bashforth scheme [*Sasaki et al.*, 2012].

Simitev-Busse-Silva (SBS) code. Pseudo-spectral numerical code for the solution of the governing equations using a poloidal-toroidal representation for the velocity and magnetic fields. Unknown scalar fields are expanded in spherical harmonics in the angular variables and Chebyshev polynomials in radius. Time

stepping is implemented by a combination of an implicit Crank-Nicolson scheme for the diffusion terms and an explicit Adams-Bashforth scheme for the Coriolis and the nonlinear terms. Early versions of the code are described in *Tilgner and Busse* [1997] and *Tilgner* [1999]. The code has been extensively modified since and used for number of years after *Simitev and Busse* [2005].

UCSC code. Spherical harmonics and Chebyshev polynomials to expand the variables in the horizontal and radial directions, respectively. Semiimplicit time integration is used (Crank-Nicolson for linear terms and Adams-Bashforth for the other terms). MPI parallelization is done for the spherical harmonic modes for the linear operation, and done in the radial direction for the nonlinear operation. The code has been modified for years after *Glatzmaier* [1984].

Calypso. Calypso uses the spherical harmonic expansion on the sphere and second-order finite difference method (FDM) in the radial direction. For time integration, Crank-Nicolson Scheme is used for the diffusion terms, and second-order Adams-Bashforth scheme is used for the other terms. Vorticity equation and Poisson equation for the toroidal vorticity are used for the time integration of the fluid motion. The directions of MPI parallelization are changed among the linear calculations, Legendre transform, and Fourier transform including nonlinear calculations. OpenMP parallelization is also used for intranode parallelization [*Matsui et al*., 2014]. We use the latest development version in the present benchmark. Ver. 1.1 is available at https://geodynamics.org/cig/software/calypso/under the GNU GPL v3 license.

ETH code. ETH code is a simulation code using the spherical harmonics for the angular component and finite differences in radius. The position of the radial grid points is chosen to be the Chebyshev roots, their interval increases toward the boundaries. A seven point stencil was used for the radial finite differences. The incompressibility condition is guaranteed by the use of a toroidal/poloidal decomposition of the vector fields. A second-order predictor-corrector scheme is used for the time integration. Diffusion terms are treated implicitly, the rest is explicit. The time step is controlled both by the corrector size which is kept to be lower than a chosen level of error and by the CFL condition [see *Marti*, 2012; *Sheyko*, 2014]. The code was developed by Ashley Willis [*Willis et al*., 2007] and the current parallelization was written by Philippe Marti.

H2000. Vectors are decomposed into poloidal and toroidal components and expanded into spherical harmonics in a sphere. Finite difference is used in the radial discretization. For time integration, Crank-Nicolson scheme is used for the diffusion terms, and Adams-Bashforth scheme is used for the other terms. This code is not parallelized [*Hollerbach*, 2000].

Leeds Spherical Dynamo (LSD). Leeds code solves the Boussinesq dynamo equations by representing velocity, and magnetic field, as poloidal and toroidal scalars. It is pseudo-spectral; the variations in a sphere are expanded in spherical harmonics, and radial variations by finite differences with nonequidistant grid using Chebyshev zeros as grid points. The nonlinear terms are evaluated by the transform method. Time stepping is by a predictor-corrector method and the time step is controlled using a CFL condition and error information from the corrector step. The LSD code is parallelized using MPI in both radial and in θ directions [*Davies et al*., 2011].

PARODY. The code uses a spherical harmonic expansion in lateral directions, and finite differences in the radial direction [*Dormy et al*., 1998]. The radial mesh interval decreases in geometrical progression toward the boundaries. A three-point stencil is used for second-order derivatives and a five-point stencil for biharmonic operators (second-order accurate). Time integration involves a Crank-Nicolson scheme for diffusion terms and a second-order Adams-Bashforth scheme for other terms. To ensure numerical stability the time step is chosen as the minimum of the characteristic time of advection and the time of Alfven wave propagation in one grid.

XSHELLS. XSHELLS uses finite differences in radius and spherical harmonic expansion in the horizontal directions. The diffusion terms are treated using the Crank-Nicolson scheme, while the other terms are treated by the second-order Adams-Bashforth scheme. Nonlinear terms are computed in physical space using the SHTns spherical harmonic transform library [*Schaeffer*, 2013]. The parallelization strategy uses MPI in the radial direction with only point-to-point communications. OpenMP can be used in the radial or the angular directions for an added level of parallelism. XSHELLS is available as free software at https://bitbucket.org/nschaeff/xshells.

TITECH code. The code uses spherical harmonic expansion in the angular directions and a combined compact finite difference method in the radial direction. A poloidal/toroidal representation is used for the magnetic and velocity fields. Time stepping is implemented by a combination of a semiimplicit Crank-Nicolson scheme for the diffusion terms and an Adams-type third order predictor-corrector scheme for the other terms. Details of the code are described in *Takahashi* [2012].

SpF-MoSST. In the model, the velocity and the magnetic field are described by the poloidal and toroidal scalars. On the spherical surfaces, spherical harmonic expansions are used. In the radial direction, both compact finite difference algorithms and the Chebyshev collocation method are employed. For the momentum equation, the radial component of the velocity, the radial component of the vorticity, and the modified pressure are solved simultaneously. For the induction equation, the radial components of the magnetic field and current are solved simultaneously [*Kuang and Bloxham*, 1999; *Jiang and Kuang*, 2008].

GeoFEM. A finite element method (FEM) with trilinear hexahedral elements is used for spatial discretization in the Cartesian coordinate. For time integration, the Crank-Nicolson scheme is used for the diffusion terms, and the other terms are solved by the Adams-Bashforth scheme. Pressure and electric potential are solved to satisfy mass conservation and Coulomb gauge for magnetic vector potential, and the Fractional step scheme is used to satisfy the divergence-free condition for velocity and magnetic vector potential [*Matsui and Okuda*, 2004a].

SFEMaNS (Ver. 2.2). Finite elements (triangle elements) and Fourier approximation are used in meridian section and azimuth direction, respectively. For MPI parallelization, the FEM mesh for the meridional section is decomposed into subdomains. The Fourier transform is also parallelized by MPI [*Guermond et al*., 2007, 2009; *Ribeiro et al*., 2015].

Appendix B: Performance Benchmark Data

Here we list the elapsed time for the performance benchmark results. The spatial resolution and elapsed time for each time step in the weak-scaling test are listed in Tables B1–B3, and the elapsed time is plotted as a function of the number of processor cores in Figure B1.

Spatial resolutions for the strong scaling test and elapsed time are also listed in Tables B4–B7, and the elapsed time is plotted as a function of number of processor cores in Figure B2.

Table B1. Elapsed Time (Second/Time Step) for Weak Scaling by Codes Using Spherical Harmonic Expansion With Chebyshev Expansion

N_{Core}	N_r	ℓ_{max}	MagIC5	Rayleigh	SBS	UCSC	N_{Core}	N_r	ℓ_{max}	SPmodel
1	16	31	0.0269936	0.102745	0.124692	0.095051	2	16	42	0.16998
8	32	63	0.0714999	0.103732	0.185473	0.098328	16	32	85	0.71648
64	64	127	0.109334	0.108528	0.138958	0.18348	128	64	170	8.27447
512	128	255	0.317476	0.131845						
4096	256	511		0.245212						

Table B2. Elapsed Time (Second/Time Step) for Weak Scaling by Codes Using Spherical Harmonic Expansion With Radial FDM

N_{Core}	N_r	ℓ_{max}	Calypso	ETH Code	LSD Code	PARODY	XSHELLS
1	64	31	0.256851	0.352721	0.248616		0.035781
8	128	63	0.465385	0.531854	0.547296	0.125999	0.051641
64	256	127	0.75803	1.71907	1.05789	0.236955	0.12273
512	512	255	0.82891	2.78505	2.13269	0.619173	0.17119
4096	1024	511	1.09464		4.00913	4.71198	0.28869

Table B3. Elapsed Time (Second/Time Step) for Weak Scaling by Codes Using Spherical Harmonic Expansion With Radial Compact FDM and Using Local Methods

					GeoFEM			SFEMaNS		
N_{Core}	N_r	ℓ_{max}	SpF-MoSST Elapsed Time	TITECH Code Elapsed Time	$\sqrt{N_{sph}}$	N_r	Time	$\sqrt{N_{med}}$	N_ϕ	Time
8	64	63	0.536304	0.361867	44.11	17	1.27525			
64	128	127	1.61485	0.72987	88.19	33	3.29256	87.29	128	0.767666
512	256	255		1.04298	176.37	65	6.31781	175.12	256	0.865461
4096					352.73	129	16.8076	345.82	512	1.07231

Table B4. Elapsed Time (Second/Time Step) for Strong Scaling by Codes Using Spherical Harmonic Expansion With Chebyshev Expansion[a]

	MagIC5	Rayleigh	SBS	UCSC	SPmodel
ℓ_{max}	255	255	256	255	170
(N_r, N_θ, N_ϕ)	(192, 384, 768)	(192, 384, 768)	(193, 385, 769)	(257, 384, 768)	(96, 256, 512)
N_{Core}	Elapsed Time				
8	9.21265				11.3106
16	6.93438		34.8506	7.8559	9.19364
32	3.65146	3.89052	18.8855	4.4581	8.79569
64	1.94200	1.80874	11.5250	3.4032	8.66073
128	1.09250	0.907353	20.3100	1.0696	8.25554
256	0.713037	0.53342			8.5006
512	0.597079	0.255983			
1024		0.133309			
2048		0.110573			
4096		0.0442248			
8192		0.0354416			

[a]ℓ_{max}: truncation of the spherical harmonics; (N_r, N_θ, N_ϕ): number of grids in (r, θ, ϕ) direction.

Table B5. Elapsed Time (Second/Time Step) for Strong Scaling by Codes Using Spherical Harmonic Expansion With Radial FDM[a]

	Calypso (HB)	Calypso (MPI)	ETH Code	LSD Code	PARODY	XSHELLS
ℓ_{max}			255			
(N_r, N_θ, N_ϕ)			(512, 384, 768)			
N_{Core}			Elapsed Time			
16					12.5429	4.0731
32					6.36380	2.07441
64			20.3963	15.1231	3.20937	1.0741
128	3.73266		10.4146	7.77790	1.68538	0.56204
256	1.71351	1.81378	5.49180	4.08026	0.951109	0.30811
512	0.828911	0.84722	2.78505	2.13269	0.619173	0.17119
1024	0.443448	0.40504		1.34670	0.564325	0.085412
2048	0.220731	0.24759		0.852113	0.575268	0.054671
4096	0.113289	0.11534		0.374556		0.039696
8192	0.060780	0.05254		0.608120		
16,384	0.034834	0.04252		0.623616		

[a]ℓ_{max}: truncation of the spherical harmonics; (N_r, N_θ, N_ϕ): number of grids in (r, θ, ϕ) direction; (HB): hybrid parallelization (MPI + OpenMP); (MPI): MPI parallelization only.

Table B6. Elapsed Time (Second/Time Step) for Strong Scaling by Codes Using Spherical Harmonic Expansion With Radial Compact FDM[a]

	SpF-MoSST	TITECH Code
ℓ_{max}	63	255
(N_r, N_θ, N_ϕ)	(124, 96, 192)	(256, 384, 768)
N_{Core}	Elapsed Time	
4	4.49747	
8	2.23190	
16	1.38402	
32	0.765525	
64	0.500593	6.40703
125	0.320104	
128		3.54131
256	1.71351	1.86101
512	0.828911	1.04298

[a]ℓ_{max}: truncation of the spherical harmonics; (N_r, N_θ, N_ϕ): number of grids in (r, θ, ϕ) direction.

Table B7. Elapsed Time (Second/Time Step) for Strong Scaling by Codes Using Local Methods[a]

	GeoFEM $(N_r, \sqrt{N_{sphere}})$	SFEMaNS $(\sqrt{N_{med}}, N_\phi)$
	$=(65, 176.4)$	$=(256.1, 128)$
N_{Core}	Elapsed Time	
64	140.32	
125	61.0636	
128	26.2981	
256	12.0594	
512	6.31781	4.55822
1024	3.61803	2.30393
2048	2.24999	1.40748
4096		0.954898
8192		0.70413
16,384		0.876519

[a]N_{sphere}: number of node on a sphere; N_{med}: number of node on a meridional plane.

Appendix C: Accuracy Benchmark Results

Accuracy tests were performed in the participants' own computer environment. We measured accuracy in terms of deviation from suggested solutions to each benchmark problem. The accuracy benchmark solutions for the insulating boundary case using SH-CHY, SH-CFDM, SH-FDM, and local methods are listed in Tables C1–C3 and

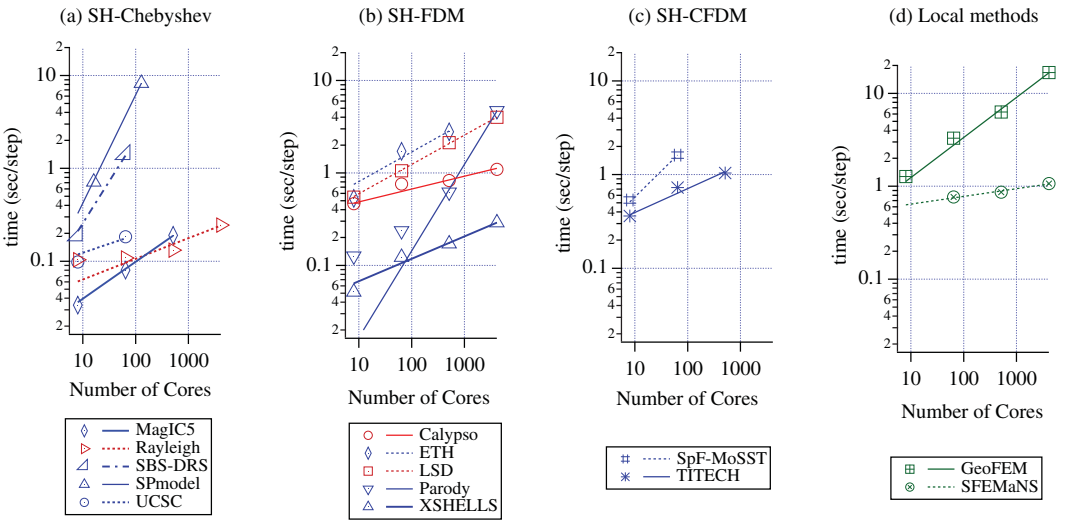

Figure B1. Elapsed time for each time step in the weak-scaling test up to 4096 processor cores. Results for H2000 are excluded because H2000 is not parallelized. Codes with 1-D parallelization are plotted in blue, codes with 2-D parallelization are plotted in red, and 3-D-parallelized models are plotted in green. A solid line is used for codes with hybrid parallelization (MPI and OpenMP), and a dotted line is used for the other codes.

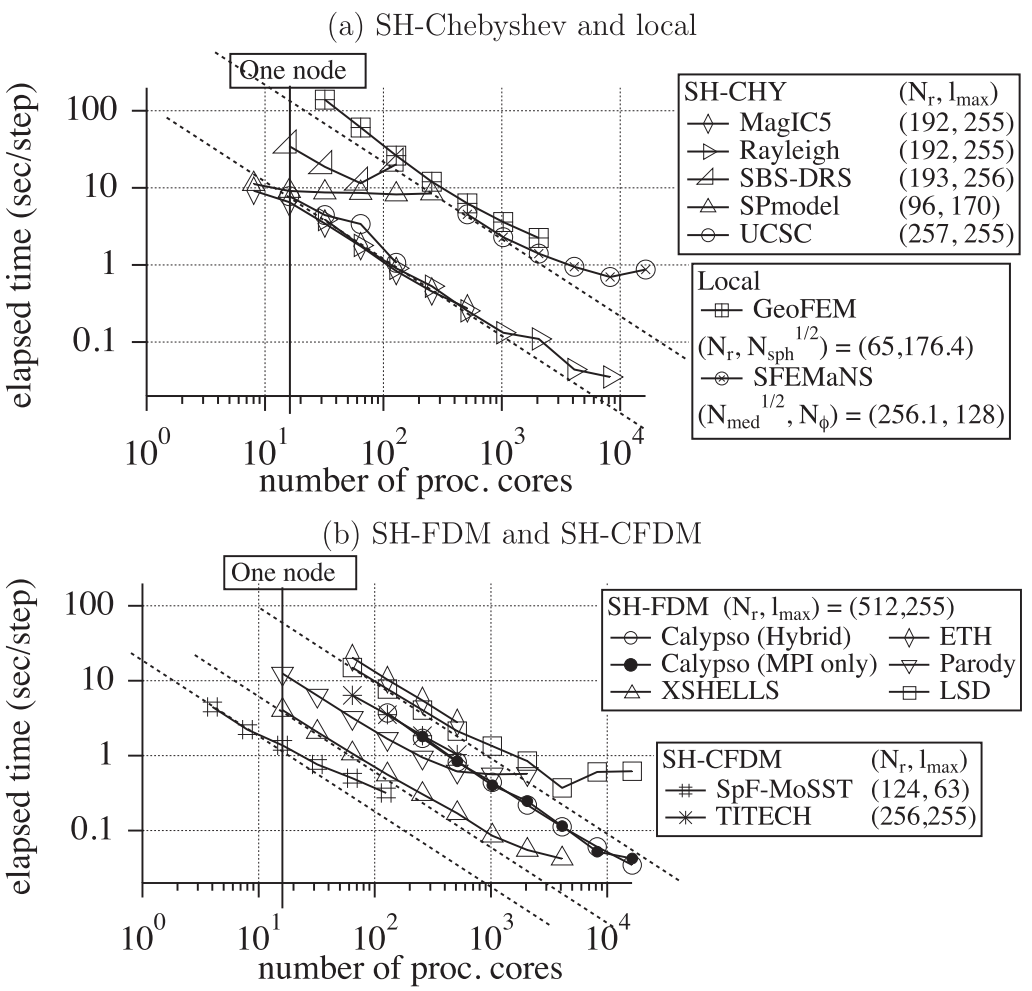

Figure B2. Elapsed time for each time step in the strong scaling as a function of the number of processor cores. The ideal scaling ($t_{elaps} \propto N_{Core}^{-1}$) for SFEMaNS, Rayleigh, LSD code, XSHELLS, and SpF-MoSST is plotted by dotted lines. Results of Calypso using hybrid (MPI and OpenMP) parallelization and MPI parallelization are plotted by open and solid circles, respectively.

C4, respectively. The suggested solutions are also listed in the top row of Tables C1–C4. Only 8 groups reported results for the pseudo-insulating boundary case, while 15 groups reported results for the insulating boundary case. The solutions for the pseudo-insulating boundary case by all the groups are listed in Table C5.

Relative errors, defined as differences from the suggested solution, depend on several factors, but are particularly sensitive to the spatial resolution. Because of this sensitivity, we organize the results of this test in terms of cube root of the degrees of freedom (DOFs) for one of the scalar fields, as done in *Christensen et al.* [2001, 2009]. For example, DOFs for a scalar field for the spherical transform method with triangular truncation is $\mathrm{DOFs}=N_r(\ell_{max}+1)^2$, where N_r is the number of radial grids.

The errors relative to the suggested solution as a function of the $(\mathrm{DOFs})^{1/3}$ for the electrically insulating boundary case and pseudo-insulating boundary case are shown in Figures C1 and C2, respectively. In relative terms, we find that the spherical harmonic expansion is a more accurate technique than the local methods, as previously reported by *Matsui and Okuda* [2005] and *Jackson et al.* [2014].

The more accurate the method, the less spatial resolution (DOFs) is required to obtain the same error. To compare the accuracy of the models, Figure C3 shows $(\mathrm{DOFs})^{1/3}$ for the six solutions that have less than 1% error from the suggested solutions. This result shows that spherical harmonic expansion methods need much less DOFs in one dimension, compared to local methods. Among the codes using spherical harmonics, the difference in accuracy results is less significant.

Table C1. Accuracy Benchmark Results for the Insulating Boundary Case (Spherical Harmonic Expansion With Chebyshev Expansion)

Code	N_r	ℓ_{max}	m_{max}	E_{kin}	E_{mag}	T	u_ϕ	B_θ	ω
Solution[a]				30.773	626.41	0.37338	−7.625	−4.9289	−3.1017
MAGIC	31	21	21	30.989	660.895	0.364179	−6.92434	−5.13508	−2.91776
MAGIC	31	32	32	30.7225	626.574	0.364957	−7.3101	−5.04949	−2.92483
MAGIC	31	42	42	30.7697	626.409	0.373843	−7.64128	−4.93392	−3.10116
MAGIC	31	64	64	30.7743	626.409	0.373416	−7.62614	−4.92877	−3.10157
MAGIC	31	85	85	30.7743	626.41	0.37339	−7.6253	−4.92892	−3.10167
MAGIC	15	64	64	30.4416	620.688	0.373061	−7.64412	−4.8762	−3.07104
MAGIC	23	64	64	30.7752	626.415	0.373417	−7.62643	−4.92879	−3.10182
MAGIC	31	64	64	30.7743	626.409	0.373416	−7.62614	−4.92877	−3.10157
MAGIC	39	64	64	30.7741	626.404	0.373416	−7.62614	−4.92874	−3.10154
Rayleigh	15	63	63	30.8305	626.983	0.37178	−7.6935	−4.88434	−3.11914
Rayleigh	21	63	63	30.778	626.332	0.37233	−7.6557	−4.89865	−3.10265
Rayleigh	31	63	63	30.774	626.410	0.37271	−7.6453	−4.90781	−3.10183
Rayleigh	42	63	63	30.774	626.364	0.37284	−7.6429	−4.91065	−3.10173
Rayleigh	63	63	63	30.774	626.386	0.37294	−7.6403	−4.91327	−3.10214
Rayleigh	85	63	63	30.774	626.394	0.37297	−7.6394	−4.91417	−3.10182
SBS	33	64	64	30.7246	625.702	0.3727	−7.6724	−4.8971	−3.1046
SPmodel	21	42	42	30.7681	626.356	0.373785	−7.63999	−4.93412	−3.10066
UCSC	25	63	31	30.684	626.002	0.37366	−7.6155	−4.8871	−3.0857

[a]Suggested solution by *Christensen et al.* [2001, 2009].

Table C2. Accuracy Benchmark Results for the Insulating Boundary Case (Spherical Harmonic Expansion With Compact FDM)

Code	N_r	ℓ_{max}	m_{max}	E_{kin}	E_{mag}	T	u_ϕ	B_θ	ω
Solution[a]				30.773	626.41	0.37338	−7.625	−4.9289	−3.1017
SpF-MoSST	34	42	32	30.739	626.26	0.359	−7.636	−4.911	−3.113
TITECH code	40	31	31	30.616	622.49	0.37404	−7.6175	−4.8646	−3.0875
TITECH code	80	31	31	30.669	624.88	0.37392	−7.5855	−4.8742	−3.0844
TITECH code	120	31	31	30.675	625.06	0.37391	−7.5848	−4.8748	−3.0847
TITECH code	30	47	47	30.534	620.5	0.37347	−7.6508	−4.9105	−3.099
TITECH code	40	47	47	30.679	624.05	0.37328	−7.6269	−4.9247	−3.0985
TITECH code	60	47	47	30.751	625.78	0.3732	−7.6178	−4.9323	−3.1005
TITECH code	80	47	47	30.765	626.15	0.37319	−7.6162	−4.9339	−3.1012

[a]Suggested solution by *Christensen et al.* [2001, 2009].

Table C3. Accuracy Benchmark Results for the Insulating Boundary Case (Spherical Harmonic Expansion With FDM)

Code	N_r	ℓ_{max}	m_{max}	E_{kin}	E_{mag}	T	u_ϕ	B_θ	ω
Solution[a]				30.773	626.41	0.37338	−7.625	−4.9289	−3.1017
Calypso	65	31	31	30.9842	632.585	0.373956	−7.63109	−4.92019	−3.12568
Calypso	72	36	31	31.0018	632.67	0.375909	−7.72544	−4.93608	−3.12835
Calypso	96	47	47	30.9394	630.91	0.373348	−7.62113	−4.95774	−3.12064
Calypso	144	47	47	30.8475	628.31	0.373289	−7.61863	−4.94572	−3.1098
H2000	120	48	48	30.774	626.41	0.37308	−7.6178	−4.9313	−3.1015
H2000	150	60	60	30.774	626.42	0.37343	−7.6267	−4.9283	−3.1017
LSD code	31	32	8	30.6999	624.616	0.3708	−7.6484	−4.834	−3.089
LSD code	31	64	16	30.8476	627.568	0.3732	−7.6274	−4.9291	−3.107
LSD code	47	64	16	30.7729	626.188	0.3732	−7.6407	−4.916	−3.1011
LSD code	63	64	16	30.763	625.987	0.3726	−7.6741	−4.892	−3.0997
LSD code	79	64	16	30.7606	625.939	0.3729	−7.6599	−4.9025	−3.1012
LSD code	95	64	16	30.7595	625.912	0.3725	−7.6609	−4.8961	−3.1008
LSD code	95	96	24	30.7649	626.19	0.3732	−7.6332	−4.9206	−3.1013
PARODY	48	44	44	30.596	616.28	0.3733	−7.769	−4.88	−3.101
PARODY	90	44	44	30.748	624.25	0.3731	−7.652	−4.925	−3.104
PARODY	150	44	44	30.767	625.71	0.3731	−7.626	−4.933	−3.103
PARODY	200	44	44	30.77	626.02	0.373	−7.62	−4.934	−3.102
XSHELLS	64	36	36	30.9012	624.308	0.37504	−7.7793	−4.8884	−3.10368
XSHELLS	48	47	47	30.8064	612.907	0.37405	−8.037	−4.7836	−3.08182
XSHELLS	96	47	47	30.8613	626.538	0.37344	−7.6719	−4.9254	−3.10863
XSHELLS	160	53	53	30.812	626.659	0.37334	−7.6373	−4.9272	−3.1052
XSHELLS	240	63	63	30.7928	626.568	0.37344	−7.6333	−4.9279	−3.10354
XSHELLS	384	74	74	30.782	626.483	0.3734	−7.6279	−4.9289	−3.10254
XSHELLS	512	85	85	30.7786	626.452	0.3734	−7.6267	−4.9289	−3.10222

[a]Suggested solution by *Christensen et al*. [2001, 2009].

Table C4. Accuracy Benchmark Results for the Insulating Boundary Case (Local Methods)[a]

Code	N_r	N_{sphere}	E_{kin}	E_{mag}	T	u_ϕ	B_θ	ω
Solution[b]			30.773	626.41	0.37338	−7.625	−4.9289	−3.1017
GeoFEM	64	7778	31.238	635.5	0.371288	−7.38088	−4.99522	−2.97504
GeoFEM	64	31106	31.404	632.98	0.373514	−7.58122	−4.99065	−3.12018
Code	N_{med}	N_ϕ	E_{kin}	E_{mag}	T	u_ϕ	B_θ	ω
SFEMaNS	13257	64	30.629	627.729	0.37362	−7.5915	−4.89809	

[a]N_r: number of points in radial direction; N_{sphere}: number of node on a sphere; N_{med}: number of node on a meridional plane; N_ϕ: number of points in ϕ direction.
[b]Suggested solution by *Christensen et al*. [2001, 2009].

Table C5. Accuracy Benchmark Results for the Pseudo-Insulating Boundary Case[a]

Code	N_r	ℓ_{max}	m_{max}	E_{kin}	E_{mag}	T	u_ϕ	B_θ	Ω
Solution[b]				40.678	219.39	0.42589	−11.636	1.4043	0.74990
SPmodel	24	31	31	40.753	218.769	0.425931	−11.6256	1.39489	0.752479
UCSC	25	53	26	40.649	219.57	0.42534	−11.636	1.4082	0.7488
SpF-MoSST	34	42	32	40.646	219.59	0.402	−11.486	1.403	0.7483
TITECH code	40	41	41	40.099	223.21	0.4253	−11.615	1.4414	0.799
TITECH code	60	47	47	40.302	222.35	0.4256	−11.645	1.4304	0.7639
TITECH code	90	47	47	40.476	220.99	0.4257	−11.641	1.4177	0.7543
Calypso	145	31	31	40.8683	218.965	0.426041	−11.6148	1.39693	0.743255
ETH code	80	42	42	40.6809	219.405	0.4259	−11.6334	1.40446	0.7502
ETH code	96	48	48	40.6798	219.405	0.4259	−11.6341	1.40502	0.74978
ETH code	96	60	60	40.6806	219.401	0.426	−11.6359	1.40417	0.7498
ETH code	128	64	64	40.6798	219.395	0.4259	−11.6357	1.40431	0.74978
ETH code	200	100	100	40.6787	219.395	0.4259	−11.6357	1.40431	0.74976
H2000	120	48	48	40.678	219.39	0.42593	−11.635	1.4039	0.74975
H2000	150	60	60	40.679	219.39	0.42595	−11.636	1.4042	0.74978
SFEMaNS	$(N_{med}, N_\phi) = (13257, 64)$			40.542	220.78	0.42553	−11.6559	1.4164	

[a]N_{med}: number of node on a meridional plane; N_ϕ: number of points in ϕ direction.
[b]Suggested solution by *Jackson et al*. [2014].

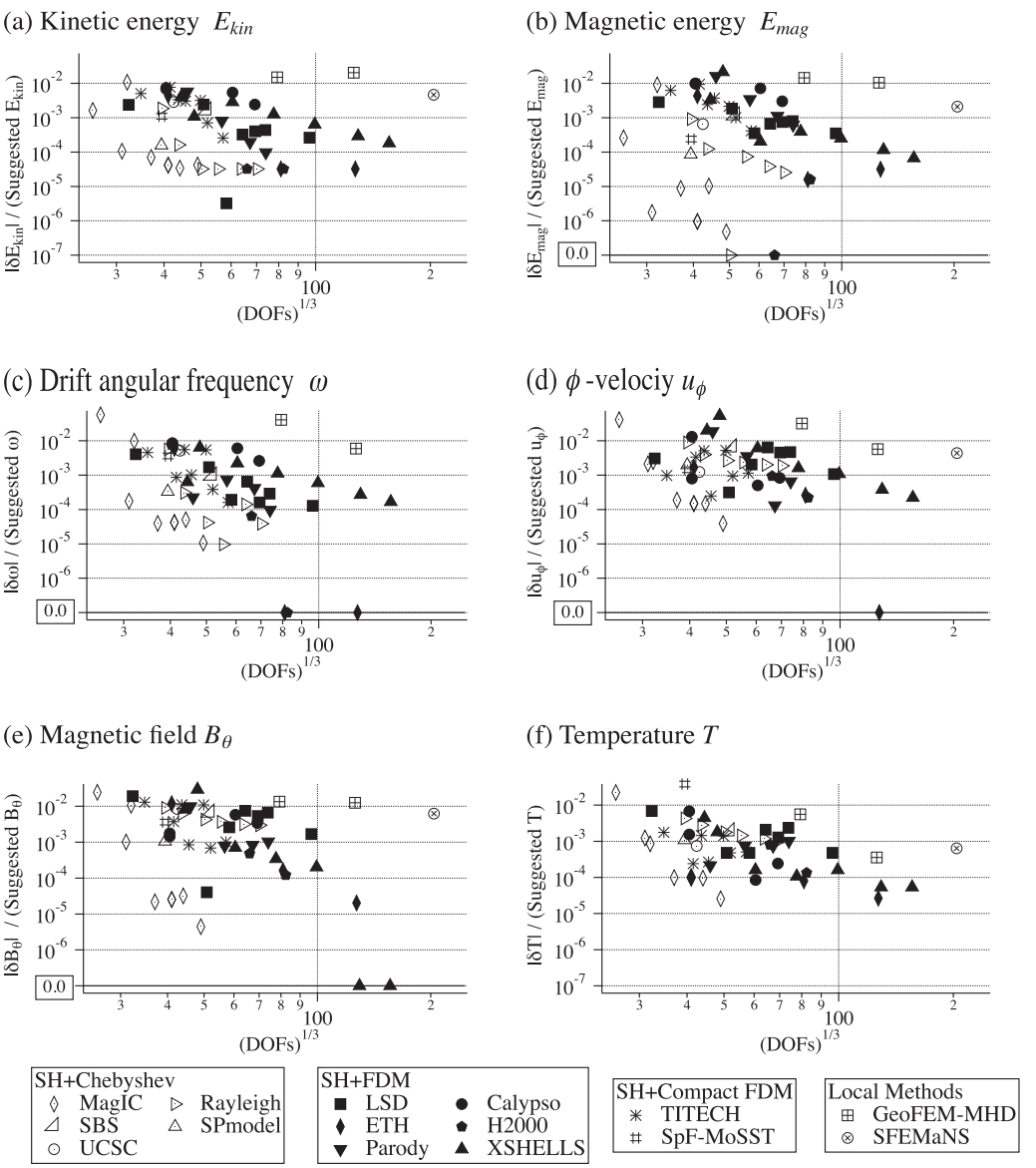

Figure C1. Relative errors of the accuracy benchmark with insulating magnetic boundary condition, as a function of the cube root of DOFs. Results with the same value as the suggested solutions are plotted at the bottom of the plot which is shown as "0.0" in the left axis. SFEMaNS and GeoFEM solutions are not fully converged because these simulations are stopped before achieving the quasi steady state.

To compare differences in the radial accuracy tolerance, we plot in Figure C4 the minimum radial resolution N_r that satisfies the 1% tolerance for the codes using spherical harmonic expansion. We find significant differences among Chebyshev expansion, compact FDM, and FDM methods in terms of accuracy defined this way. Specifically, the Chebyshev expansion method only needs one third of the radial points to match the FDM accuracy, whereas the compact FDM methods require a similar radial resolution to the Chebyshev expansion. These results are consistent with results previously reported by *Christensen et al.* [2001, 2009]. Differences in code accuracy in Figure C1 are more systematic in global functionals (i.e., E_{kin}, E_{mag}, and ω) than in local functionals, because two steps are needed to obtain the local field data; locating the measurement point and evaluating the solution at that point. Each of these introduces error.

One question that remains unanswered is how the solution converges with increasing resolution. In the present study, we performed the benchmark with $\ell_{max} < 100$ and $N_r = 200$ to obtain 1% error tolerance. To

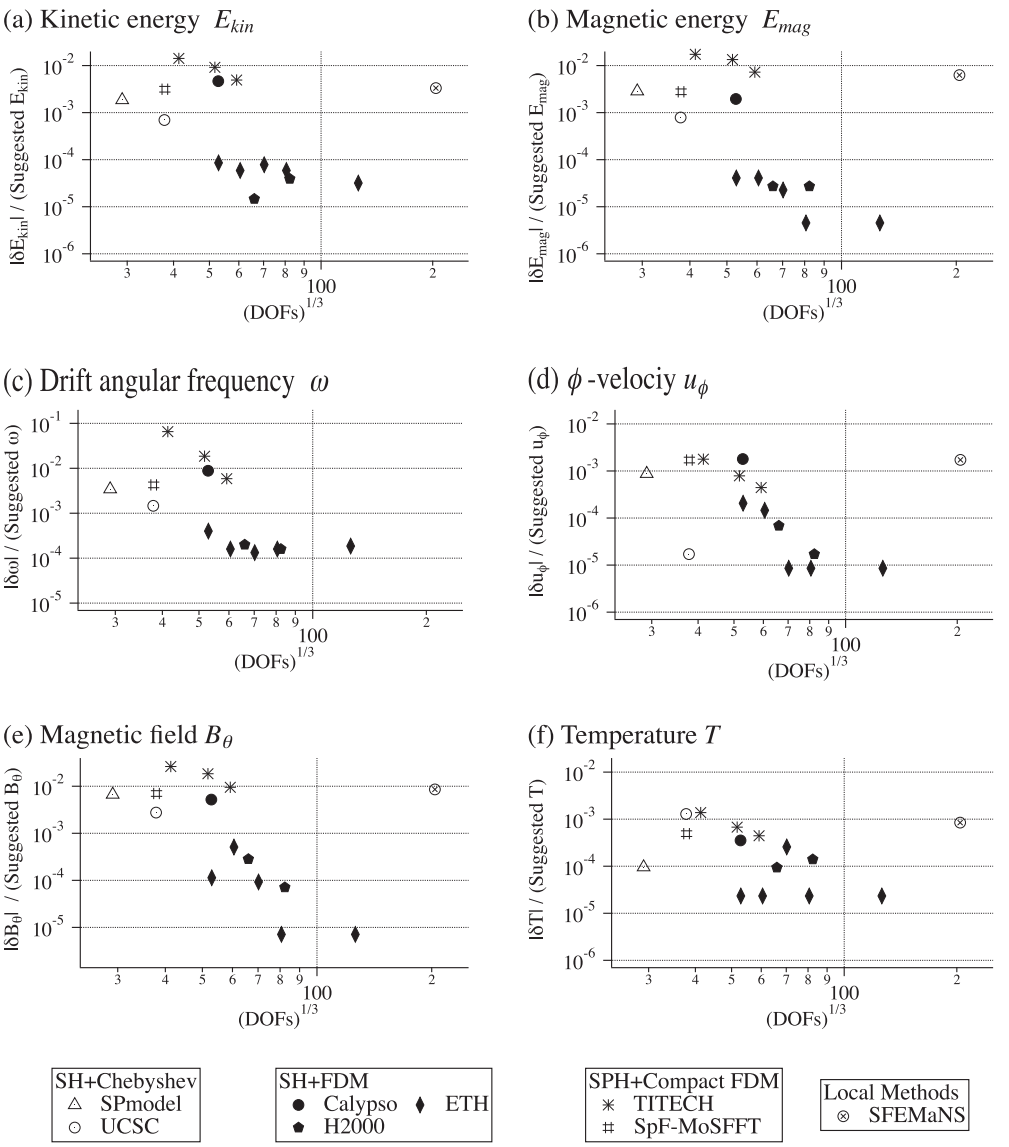

Figure C2. Relative errors of the accuracy benchmark with the pseudo-insulating magnetic boundary as a function of the cube root of DOFs. SFEMaNS solutions are not fully converged because these simulations are stopped before achieving the quasi steady state.

investigate the convergence of the solution, we need to obtain the solution with larger resolution (e.g., $\ell_{max} = 2048$ and $N_r = 1024$), and compare results with varying radial resolution while keeping the truncation of the spherical harmonics.

In summary, the Chebyshev expansion method and compact FDM also require approximately 3 times less DOFs than that for FDM in the radial direction. For the other direction, the spherical harmonic expansion method requires fewer DOFs in one direction than that for local methods to obtain the same accuracy.

It is noted that GeoFEM and SFEMaNS use unstructured grids, but we used simple grid pattern in our tests of these codes. It is possible that a more highly optimized FEM mesh based on the scale of the local flow and magnetic field patterns might allow these codes to obtain better accuracy with fewer DOFs than that in the present study.

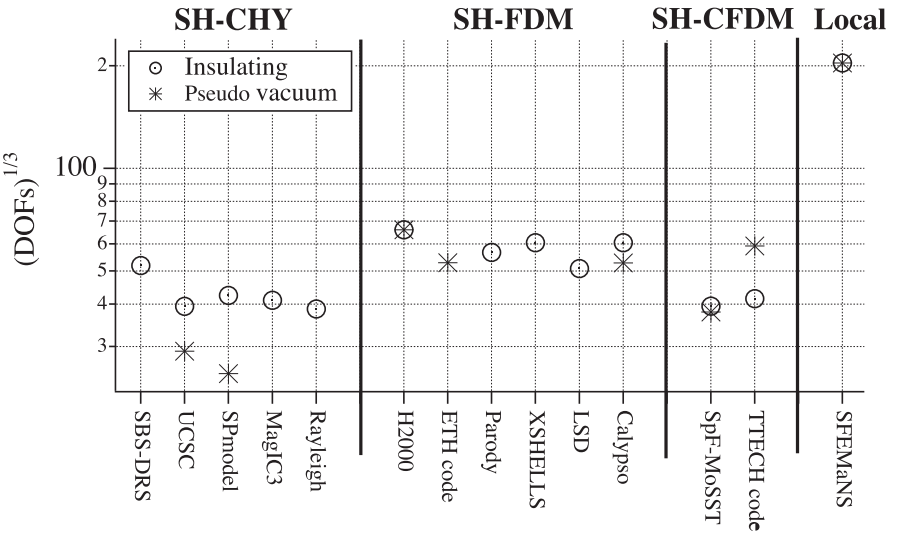

Figure C3. Cube root of DOFs needed to satisfy 1% error tolerance for all six solution variables (E_{kin}, E_{mag}, ω, T, u_ϕ, and B_θ; see section A3) in reported data of the accuracy benchmark. Results for the insulating boundary case are plotted with open circles, and for the pseudo-insulating boundary case with asterisks. Results from GeoFEM are not plotted since GeoFEM's solution does not satisfy the 1% tolerance for all the solution variables.

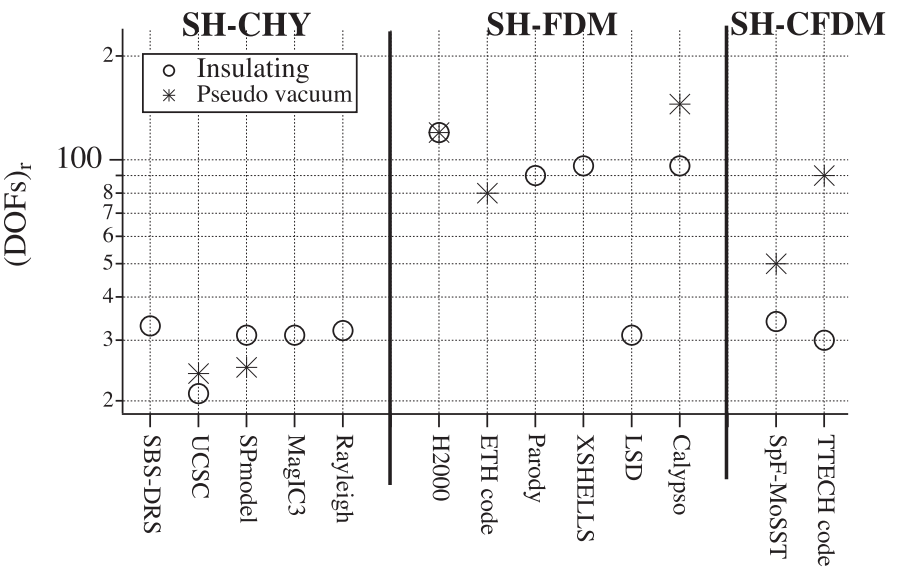

Figure C4. Number of radial grid points (expressed as radial DOFs) needed to satisfy 1% error tolerance for all six required solution variables (E_{kin}, E_{mag}, ω, T, u_ϕ, and B_θ; see section A3) for codes using spherical harmonic expansion. Results for the insulating boundary case are plotted by open circles, and those for pseudo-insulating boundary case are shown by asterisks.

Acknowledgments
This work was supported by the Computational Infrastructure for Geodynamics (CIG), which is supported by the National Science Foundation award NSF-0949446. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1053575. The Calypso and Rayleigh codes have been developed and released as open source programs under a support from CIG. We thank all the participants at the 2012 Geodynamo Developer Meeting which drove this work and the follow-on 2015 Geodynamo Benchmark Workshop. R. D. Simitev and L. Silva acknowledge support of the Leverhulme Trust Research Project grant RPG-2012-600 in the UK. We also thank the reviewers for useful suggestions.

References

Chan, K. H., K. K. Zhang, J. Zou, and G. Schubert (2001), A nonlinear, 3-D spherical α^2 dynamo using a finite element method, *Phys. Earth Planet. Inter.*, *128*, 35–50.

Chan, K. H., K. Zhang, L. Li, and X. Liao (2007), A new generation of convection-driven spherical dynamos using EBE finite element method, *Phys. Earth Planet. Inter.*, *163*, 251–265.

Christensen, U. R., and J. Wicht (2007), *Numerical Dynamo Simulations, in Treatise on Geophysics, vol. 8, Core Dynamics*, edited by P. Olson, pp. 245–282, Elsevier B. V., Amsterdam.

Christensen, U. R., et al. (2001), A numerical dynamo benchmark, *Phys. Earth Planet. Inter.*, *128*, 25–34.

Christensen, U. R., et al. (2009), Erratum to "A numerical dynamo benchmark" [Phys. Earth Planet. Int. 128(14) (2001) 2534], *Phys. Earth Planet. Inter.*, *172*, 356.

Davies, C. J., D. Gubbins, and P. K. Jimack (2011), Scalability of pseudospectral methods for geodynamo simulations, *Concurrency Comput.*, *23*, 38–56.
Dormy, E., P. Cardin, and D. Jault (1998), MHD flow in slightly differentially rotating spherical shell and with conducting inner core and in dipolar magnetic field, *Earth Planet. Sci. Lett.*, *160*, 15–30.
Engelmann, C. (2014), Scaling to a million cores and beyond: Using light-weight simulation to understand the challenges ahead on the road to exascale, *Future Gen. Comput. Syst.*, *30*, 59–65.
Featherstone, N., and B. W. Hindman (2016), The spectral amplitude of stellar convection and its scaling in the high-Rayleigh-number regime, *Astrophys. J.*, *818*, 32, doi:10.3847/0004-637X/818/1/32.
Glatzmaier, G. A. (1984), Numerical simulations of stellar convective dynamos. I. The model and method, *J. Comput. Phys.*, *55*, 461–484.
Glatzmaier, G. A., and P. H. Roberts (1995), A three-dimensional self-consistent computer simulation of a geomagnetic field reversal, *Nature*, *377*, 203–209.
Guermond, J.-L., R. Laguerre, J. Léorat, and C. Nore (2007), An interior penalty Galerkin method for the MHD equations in heterogeneous domains, *J. Comput. Phys.*, *221*, 349–369.
Guermond, J.-L., R. Laguerre, J. Léorat, and C. Nore (2009), Nonlinear magnetohydrodynamics in axisymmetric heterogeneous domains using a Fourier/finite element technique and an interior penalty method, *J. Comput. Phys.*, *228*, 2739–2757.
Harder, H. and U. Hansen (2005), A finite-volume solution method for thermal convection and dynamo problems in spherical shells, *Geophys. J. Int.*, 522–532.
Hollerbach, R. (2000), A spectral solution of the magneto-convection equations in spherical geometry, *Int. J. Numer. Methods Fluids*, *32*, 773–797.
Hotta, H., M. Rempel, and T. Yokoyama (2016), Large-scale magnetic fields at high Reynolds numbers in magnetohydrodynamic simulations, *Science*, *351*, 1427–1430.
Jackson, A., et al. (2014), A spherical shell numerical dynamo benchmark with pseudo-vacuum magnetic boundary conditions, *Geophys. J. Int.*, *196*, 712–723.
Jiang, W., and W. Kuang (2008), An MPI-based MoSST core dynamics model, *Phys. Earth Planet. Inter.*, *170*, 46–51.
Kuang, W., and J. Bloxham (1999), Numerical modeling of magnetohydrodynamic convection in a rapidly rotating spherical shell: weak and strong field dynamo action, *J. Comput. Phys.*, *153*, 51–81.
Marti, P. (2012), Convection and boundary driven flows in a sphere, PhD thesis, ETH-Zürich, Zürich, Switzerland, doi:10.3929/ethz-a-007329939.
Matsui, H., and H. Okuda (2004a), Development of a simulation code for MHD dynamo processes using the GeoFEM platform, *Int. J. Comput. Fluid Dyn.*, *18*, 323–332.
Matsui, H., and H. Okuda (2004b), Treatment of the magnetic field for geodynamo simulations using the finite element method, *Earth Planets Space*, *56*, 945–954.
Matsui, H., and H. Okuda (2005), MHD dynamo simulation using the GeoFEM platform—Verification by the dynamo benchmark test, *Int. J. Comput. Fluid Dyn.*, *19*, 15–22.
Matsui, H., E. King, and B. A. Buffett (2014), Multi-scale convection in a geodynamo simulation with uniform heat flux along the outer boundary, *Geochem. Geophys. Geosyst.*, *15*, 3212–3225, doi:10.1002/2014GC005432.
Orszag, S. A., and G. S. Patterson Jr. (1972), Numerical simulation of turbulence, in *Statistical Models and Turbulence*, edited by M. Rosenblatt and C. Van Atta, pp. 127–147, Springer, Berlin, Heidelberg.
Ribeiro, A., G. Fabre, J.-L. Guermond, and J. M. Aurnou (2015), Canonical models of geophysical and astrophysical flows: Turbulent convection experiments in liquid metals, *Metals*, *5*, 289–335.
Sasaki, Y., S. Takehiro, Y.-Y. Hayashi, and SPMODEL Development Group (2012), *Project of MHD Dynamo in Rotating Spheres and Spherical Shells*, Kyoto, Japan. [Available at http://www.gfd-dennou.org/library/dynamo/.]
Schaeffer, N. (2013), Efficient spherical harmonic transforms aimed at pseudospectral numerical simulations, *Geochem. Geophys. Geosyst.*, *14*, 751–758, doi:10.1002/ggge.20071.
Sheyko, A. (2014), *Numerical Investigations of Rotating MHD in a Spherical Shell*, PhD thesis, ETH-Zürich, Zürich, Switzerland, doi:10.3929/ethz-a-010198511.
Simitev, R. D., and F. H. Busse (2005), Prandtl number dependence of convection driven dynamos in rotating spherical fluid shells, *J. Fluid Mech.*, *532*, 365–388.
Stanley, S., and G. A. Glatzmaier (2010), Dynamo models for planets other than Earth, *Space Sci. Rev.*, *152*, 617–649.
Takahashi, F. (2012), Implementation of a high-order combined compact difference scheme in problems of thermally driven convection and dynamo in rotating spherical shells, *Geophys. Astrophys. Fluid Dyn.*, *106*, 231–249.
Texas Advanced Computing Center (2013), *TACC Stampede User Guide*, Austin, Tex. [Available at https://portal.tacc.utexas.edu/user-guides/stampede.]
Tilgner, A. (1999), Spectral methods for the simulation of incompressible flows in spherical shells, *Int. J. Numer. Methods Fluids*, *30*, 713–724.
Tilgner, A., and F. H. Busse (1997), Finite-amplitude convection in rotating spherical fluid shells, *J. Fluid Mech.*, *332*, 359–376.
Wicht, J., S. Stellmach, and H. Harder (2009), Numerical models of the geodynamo: From fundamental Cartesian models to 3d simulations of field reversals, in *Geomagnetic Field Variations, Adv. Geophys. Environ. Mech. Math.*, pp. 107–158, Springer, Berlin.
Willis, A. P., B. Sreenivasan, and D. Gubbins (2007), Thermal core–mantle interaction: Exploring regimes for 'locked' dynamo action, *Phys. Earth Planet. Inter.*, *165*, 83–92.